

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-306879

(43)Date of publication of application : 21.11.1995

(51)Int.CI.

G06F 17/50

(21)Application number : 06-097093

(71)Applicant : NEC CORP

(22)Date of filing : 11.05.1994

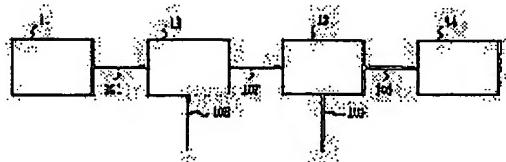
(72)Inventor : ASAOKI YOSHI

## (54) NET LIST HARDWARE DESCRIPTION CONVERSION DEVICE

## (57)Abstract:

PURPOSE: To make a shift to function level design in hardware description language by generating a hardware description language source from a net list.

CONSTITUTION: This device is equipped with a net list file 11 which contains a net list of gate level, a register transfer information generating means 12 which reads the gate level net list 101 out of the net list file 11 and generates and outputs register transfer information 103 by referring to element operation information 102 prescribing the operation of a circuit element included in the gate level net list 101, and a hardware description output means 13 which inputs the register transfer information 103, generates and outputs the hardware description language source 105 by using a description output template 104, and stores it in a source file 14.



## LEGAL STATUS

[Date of request for examination] 11.05.1994

[Date of sending the examiner's decision of rejection] 06.05.1997

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (JP)

## (12)公開特許公報 (A)

(11)特許出願公開番号

特開平7-306879

(43)公開日 平成7年(1995)11月21日

(51)Int.Cl.  
G06F 17/50

識別記号

F I

7623-5L

G06F 15/60

360

K

審査請求 有 請求項の数4 O L (全6頁)

(21)出願番号 特願平6-97093

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(22)出願日 平成6年(1994)5月11日

(72)発明者 浅尾 清

東京都港区芝五丁目7番1号 日本電気株式会社内

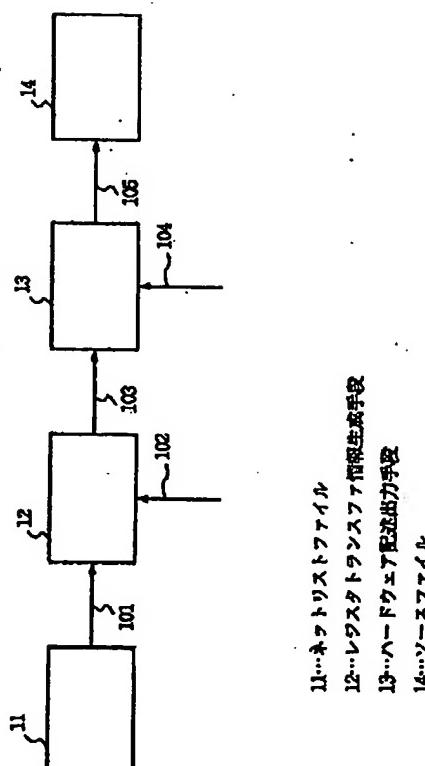
(74)代理人 弁理士 京本 直樹 (外2名)

(54)【発明の名称】ネットリスト・ハードウェア記述変換装置

## (57)【要約】

【目的】ネットリストからハードウェア記述言語ソースを生成し、ハードウェア記述言語による機能レベル設計への移行を図る。

【構成】本発明のネットリスト・ハードウェア記述変換装置は、ゲートレベルのネットリストを収納するネットリスト・ファイル11と、ネットリストファイル11よりゲートレベル・ネットリスト101を読み取り、ゲートレベル・ネットリスト101に含まれている回路素子の動作を規定する素子動作情報102を参照して、レジスタトランスマニア情報103を生成して出力するレジスタトランスマニア情報生成手段12と、このレジスタトランスマニア情報103を入力し、記述出力テンプレート104を用いてハードウェア記述言語ソース105を生成して出力し、ソースファイル14に収納するハードウェア記述出力手段13とを備えて構成される。



## 【特許請求の範囲】

【請求項1】 素子間接続情報を収納するネットリスト・ファイルよりネットリストを読み出し、当該ネットリストに含まれる素子の動作機能を定義する所定の素子動作情報を用いて、レジスタトランスマニア情報を生成して出力するレジスタトランスマニア情報生成手段と、前記レジスタトランスマニア情報を入力して、所定の記述出力テンプレートを用いてハードウェア記述言語ソースを生成して出力し、所定のソースファイルに収納するハードウェア記述出力手段と、を少なくとも備えることを特徴とするネットリスト・ハードウェア記述変換装置。

【請求項2】 前記素子情報が、NOT回路、AND回路、NAND回路、OR回路、NOR回路、EXOR回路およびEXNOR回路等を含む素子に対しては、これらの各素子の機能として適用される論理式により規定されることを特徴とする請求項1記載のネットリスト・ハードウェア記述変換装置。

【請求項3】 前記素子情報が、少なくともD型フリップフロップを含む素子に対しては、当該素子の機能として適用されるデータの取り込み、非同期クリアおよび非同期プリセット等を含む動作仕様を規定する情報であることを特徴とする請求項1記載のネットリスト・ハードウェア記述変換装置。

【請求項4】 前記レジスタトランスマニア情報を、少なくともD型フリップフロップを含む素子に対しては、当該素子の機能として適用される前記動作仕様に準拠して、当該素子の周辺回路により決められる動作条件を規定することを特徴とする請求項1記載のネットリスト・ハードウェア記述変換装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明はネットリスト・ハードウェア記述変換装置に関し、特にゲートレベルのネットリストよりハードウェア設計データとして利用するためのハードウェア記述言語を生成するネットリスト・ハードウェア記述変換装置に関する。

## 【0002】

【従来の技術】近年における半導体集積回路における要求仕様の複雑さおよび回路設計規模の増大に伴ない、当該回路設計の傾向としては、従来のゲートレベル設計からハードウェア記述言語による機能レベル設計に移行しつつある。

【0003】従来、ハードウェア記述言語により定義されたハードウェア仕様より、ゲートレベルの論理回路を合成する装置については、例えば、特開平2-41572号公報において論理回路合成方式が提案されており、また他方において、特開平3-84676号公報においては、論理回路の自動合成装置が提案されている。

【0004】これらの場合の内容は、太盛田と申す。

異にしており、これらの提案における基本的なハードウェア設計手法としては、レジスタトランスマニアレベルのハードウェア記述言語または機能ブロック図、或はまた回路合成ルール指定記述等を含む入力仕様に準拠して、機能ブロックの合成ルールに従って詳細な論理回路等を合成する方法が用いられており、当該合成結果による論理回路はネットリストとして出力されている。しかしながら、このように、従来行われている機能から論理に変換するという装置とは逆に、論理から機能に変換する装置は、現時点においては見当らないのが実状である。

## 【0005】

【発明が解決しようとする課題】前述したように、近年の半導体集積回路に対する要求仕様の複雑さならびに回路規模の増大に伴い、従来の論理（ゲートレベル）設計からハードウェア記述言語による機能レベルの設計に重点が置かれつつある。この機能設計は、ハードウェア記述言語により回路の動作仕様を定義することにより行われており、当該機能設計の検証は機能レベルのシミュレータを用いて行われている。そして機能設計完了後においては、当該機能設計は、論理合成装置を用いて求める論理回路に変換されている。このような設計手法においては、新規に論理回路等を設計する場合には問題ないが、設計の対象とする論理回路に、ゲートレベルの論理回路図作成により設計された既設計データによる論理回路が混在する場合には、即ち、既設計回路をライブラリ／マクロとして使用する場合、または既設計回路の一部を流用する場合などにおいては、設計結果に対する設計検証において問題が生じている。

【0006】即ち、上述のハードウェア記述言語を入力とする機能レベルシミュレータがゲートレベルのシミュレータではないことに起因して、既設計論理回路に対してはシミュレーションすることができないという問題があり、これにより、従来使用されているネットリスト・ハードウェア記述変換装置においては、設計者として、前記機能レベルでの設計作業に移行することが困難であるという欠点がある。

## 【0007】

【課題を解決するための手段】本発明のネットリスト・ハードウェア記述変換装置は、素子間接続情報を収納するネットリスト・ファイルよりネットリストを読み出し、当該ネットリストに含まれる素子の動作機能を定義する所定の素子動作情報を用いて、レジスタトランスマニア情報を生成して出力するレジスタトランスマニア情報生成手段と、前記レジスタトランスマニア情報を入力して、所定の記述出力テンプレートを用いてハードウェア記述言語ソースを生成して出力し、所定のソースファイルに収納するハードウェア記述出力手段とを少なくとも備えて構成される。

【0008】なお、前記素子情報は、NOT回路、AND回路、NAND回路、OR回路、NOR回路、EXOR

R回路およびEX-NOR回路等を含む素子に対しては、これらの各素子の機能として適用される論理式により規定される情報であるものとしてもよく、或はまた、前記素子情報としては、少なくともD型フリップフロップを含む素子に対しては、当該素子の機能として適用されるデータの取り込み、非同期クリアおよび非同期プリセット等を含む動作仕様を規定する情報であるものとしてもよい。

【0009】また、前記レジスタトランスマニア情報は、少なくともD型フリップフロップを含む素子に対しては、当該素子の機能として適用される前記動作仕様に準拠して、当該素子の周辺回路により決められる動作条件を規定する情報であるものとしてもよい。

#### 【0010】

【実施例】次に、本発明について図面を参照して説明する。

【0011】図1は、本発明の一実施例を示すシステム構成図であり、ゲートレベルの論理回路図作成により設計された既設計データによる論理回路が混在する場合における一実施例を示している。図1に示されるように、本実施例は、既設計データであるゲートレベルのネットリストを収納するネットリスト・ファイル11と、ネットリストファイル11よりゲートレベル・ネットリスト101を読み取り、ゲートレベル・ネットリスト101に含まれている回路素子の動作情報を用いてレジスタトランスマニア情報103を生成して出力するレジスタトランスマニア情報生成手段12と、このレジスタトランスマニア情報103を受けて、ハードウェア記述言語ソース105を生成して出力するハードウェア記述出力手段13と、ハードウェア記述言語ソース105を収納するソースファイル14とを備えて構成される。

【0012】図1において、ネットリストファイル11には、既設計のゲートレベル・ネットリスト101が収納されており、このゲートレベル・ネットリスト101には、既設計のゲートレベル設計の結果によるゲート素子間の接続情報であるゲートレベル・ネットリスト、即ち回路図と等価な情報が含まれている。レジスタトランスマニア情報生成手段12においては、このゲートレベル・ネットリスト101を読み込み、このゲートレベル・ネットリスト101に現われる回路素子の動作を定義している素子動作情報102を参照して、レジスタトランスマニア情報103を生成して出力する。素子動作情報102には、ゲートであれば端子名と論理式を生成するための論理情報が定義されており、フリップフロップのように値を保持する素子の場合には、端子名と保持される値とが端子の値の変化により、どのように変わるかという情報が定義されている。例えば、2入力AND回路の場合には、入力端子A、Bと出力端子Yがあり、出力端子Yは、入力端子AおよびBの演算結果であることが定義されている。また、D型フリップフロップの場合に

は、入力端子D、CLK、PRE、CLRおよび出力端子Qがあり、CLKの立ち上がりでDの値が取り込まれて保持され、またPREの値が0の時には保持される値が1になり、CLRの値が0の時には保持される値が0になることと、保持されている値がQに出力されることなどが定義される。

【0013】レジスタトランスマニア情報生成手段12においては、まずネットリストの素子間の接続を参照して、フリップフロップのように値を保持する素子の全て

10 の入力端子の論理式が組み立てられる。この論理式の項は、回路の外部端子とフリップフロップのような値を保持する素子の出力端子である。この論理式生成のために素子動作情報102が利用される。次に、フリップフロップのように値を保持する素子については、対応する素子動作情報102により、当該素子の入力端子の値の変化に対応して、どのような値が保持されるかがわかるので、入力端子の論理式を用いて、素子に対する条件とその条件が成立したときの動作状態を示す表が作成されて、レジスタトランスマニア情報103として、レジスタトランスマニア情報生成手段12より出力される。

【0014】ハードウェア記述出力手段13においては、レジスタトランスマニア情報生成手段12より出力されるレジスタトランスマニア情報103の入力を受けて、記述出力テンプレート104を参照して、当該レジスタトランスマニア情報104より、ハードウェア記述言語の文法に適合した形式によるハードウェア記述言語ソース105が生成されて出力され、ソースファイル14に収納される。上述の記述出力テンプレート104には、論理式の表記法およびフリップフロップなどの動作記述の

20 テンプレートが定義されている。例えば、AND演算には $\neq$ 、OR演算には+というように表記するとか、D型フリップフロップであれば、if <クリア条件> then <インスタンス名> := 0 else if <データ取り込み条件> then <インスタンス名> := <取り込みデータ> endというように定義される。

【0015】ハードウェア記述出力手段13においては、記述出力テンプレート104の空欄にレジスタトランスマニア情報103より得られた情報を埋め込むことにより、ハードウェア記述言語ソース105が生成され

30 て、ソースファイル14に出力され収納される。なお、記述出力テンプレート104の内容を変えることにより、様々な種類のハードウェア記述言語ソースに対応することができる。

【0016】以上の動作を回路例を用いて示すと、以下のようになる。

【0017】図2は、既設計データの回路図の一部を示す図であり、D型フリップフロップ27(G<sub>1</sub>)と、その周辺のD型フリップフロップ21(G<sub>1</sub>)、AND回路22(G<sub>1</sub>)、AND回路24(G<sub>1</sub>)およびAND回路28(G<sub>1</sub>)、OR回路23(G<sub>1</sub>)、NAND回

路25 (G<sub>1</sub>) およびNOR回路26 (G<sub>1</sub>) 等を含む回路構成が示されている。これらの素子間の接続情報は、ネットリストファイル11に収納されている既設計のゲートレベル・ネットリスト101そのものである。レジスタトランスマニア情報生成手段12においては、まず、D型フリップフロップ27 (インスタンス名: G<sub>1</sub>) の入力端子に対応する論理式が組み上げられる。下記の表1は、この結果による端子論理式を示す表であり、当該論理式の表記としては、AND演算は・で表わし、OR演算は+で表わしており、NOT演算はバーを用いて表わしている。

【0018】

【表1】

D-FF インスタンス G<sub>4</sub> 端子論理式

端子	論理式
PRE	$\overline{T_4} \cdot T_5$
CLR	$T_6 + \overline{T_7}$
D	$(G_0 + T_0) + T_1$
CLK	$T_2 \cdot T_3$

【0019】次いで、レジスタトランスマニア情報生成手段12においては、D型フリップフロップ27 (インスタンス名 G<sub>1</sub>) の動作を表わすレジスタトランスマニア情報103が作成される。下記の表2は、このレジスタトランスマニア情報103の作成結果を示す図であり、表2においては、立ち上がりエッジはrise (……) で示されている。

【0020】

【表2】

D-FF インスタンス G<sub>4</sub> レジスタトランスマニア情報

条件	動作
$T_4 \cdot T_5 = 0$	プリセット
$T_6 + \overline{T_7} = 0$	クリア
rise ( $T_2 \cdot T_3$ )	データ取り込み 取り込みデータは $(G_0 + T_0) + T_1$

【0021】また、ハードウェア記述出力手段13においては、一例として下記の表3に示されるようなD-F (D型フリップフロップ) および論理式の記述出力テンプレートを含む記述出力テンプレート104を用いて、上記の表2に示されるレジスタトランスマニア情報103より、ハードウェア記述言語ソース105が生成されてを出力される。この場合におけるハードウェア記述言語ソース105が、表4に示される。

【0022】

【表3】

## D-FF 記述出力テンプレート

```

if <プリセット条件> then
  <インスタンス名> :=1;
else if <クリア条件> then
  <インスタンス名> :=0;
else if <データ取り込み条件> then
  <インスタンス名> := <取り込みデータ>;
end;

```

## 論理式 記述出力テンプレート

レジスタトランスマニア情報の表記	ハードウェア記述言語での表記
<L>・<R>	<L> and <R>
<L>+<R>	<L> or <R>
<E>	not <E>
rise(<E>)	<E> · up

## 【0023】

## 【表4】

```

if not(T4 and T5)=0 then
  G4 :=1;
else if not(T6 or T7)=0 then
  G6 :=0;
else if (T2 and T3) · up then
  G4 :=(G6 and T6) or T1;
end;

```

## 【0024】

【発明の効果】以上説明したように、本発明は、ゲートレベル・ネットリストからレジスタトランスマニア情報を生成し、当該レジスタトランスマニア情報をハードウェア記述言語によるソースファイルを生成することによって、ゲートレベル・ネットリストからハードウェア記述言語ソースへの情報変換が可能となり、これにより、設計の対象とする論理回路に、ゲートレベルの論理回路図

作成による既設計データによる論理回路が混在する場合においても、既設計論理回路に対応するシミュレーションを行うことが可能となり、設計者として、前記機能レベルでの設計作業に移行することができるという効果がある。

## 【図面の簡単な説明】

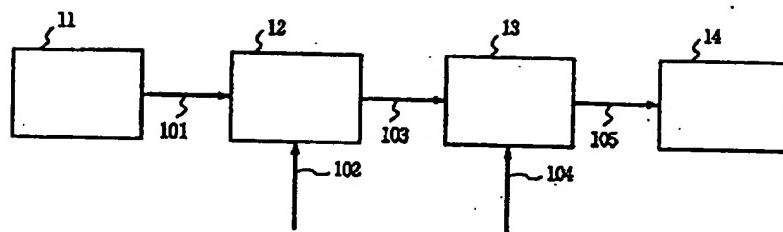
【図1】本発明の一実施例を示すシステム構成図である。

## 【図2】ネットリストの回路構成例を示す図である。

## 【符号の説明】

- 1 1 ネットリストファイル
- 1 2 レジスタトランスマニア情報生成手段
- 1 3 ハードウェア記述出力手段
- 1 4 ソースファイル
- 2 1、2 7 D型フリップフロップ
- 2 2、2 4、2 8 AND回路
- 2 3 OR回路
- 2 5 NAND回路
- 2 6 NOR回路

【図 1】



- 11…ネットリストファイル  
12…レジストランスマニア情報生成手段  
13…ハードウェア記述出力手段  
14…ソースファイル

【図 2】

